

Herramientas interactivas para la enseñanza de robótica

Carlos A. Jara, Francisco A. Candelas, Fernando Torres

Dept. de Física, Ingeniería de Sistemas y Teoría de la Señal

Universidad de Alicante

03080, San Vicente del Raspeig

cajb@dfists.ua.es, Francisco.Candelas@ua.es, Fernando.Torres@ua.es

Resumen

Este trabajo presenta herramientas interactivas que pueden ser utilizadas para la enseñanza de robótica. Mediante estas herramientas el usuario puede adquirir desde conceptos básicos de robótica hasta un aprendizaje práctico en el manejo de robots. Las aplicaciones ofrecen una interfaz y un entorno muy real que permite la práctica de varios fundamentos de robótica. También se realiza una breve descripción de *Easy Java Simulations* (Ejs), software basado en Java que se ha utilizado para la creación de las aplicaciones.

1. Introducción

La utilización de dispositivos robóticos requiere emplear a gente preparada y cualificada para el uso efectivo de estos recursos. De esta manera, una adecuada formación y adiestramiento de los estudiantes, se está convirtiendo en la llave que abre las puertas a la prosperidad dentro del campo industrial.

En la mayoría de las ocasiones, los sistemas robóticos son complejos y presentan dificultades en su manejo. Además, ofrecer una enseñanza apropiada a los estudiantes sobre estos sistemas, requiere mucho tiempo y coste. También es posible que el equipo pueda ser dañado si éste se usa inadecuadamente. Actualmente, nuevas tecnologías tales como la *realidad virtual*, permiten al operador adquirir habilidades en el uso de tales sistemas robóticos de una manera efectiva y económica. Hoy en día, se emplea frecuentemente herramientas interactivas o laboratorios virtuales en la enseñanza, ya que proporcionan un gran conocimiento práctico sobre conceptos robóticos.

En este trabajo, los autores presentan un laboratorio virtual y otro remoto para la enseñanza y adiestramiento de robótica, que permite a los usuarios interactuar con simulaciones de robots reales de una manera intuitiva y económica. Mediante las aplicaciones es posible un aprendizaje rápido de conceptos de robótica, tales como la estructura mecánica (articulaciones y eslabones), parámetros Denavit-Hartenberg, cinemática directa e inversa y planificación de trayectorias. El usuario podrá simular y teleoperar la trayectoria del robot y así, comprobar el comportamiento del mismo. Además, son muy sencillos de utilizar, incluso para la gente que no está familiarizada con el mundo de la robótica.

El artículo se ha organizado de la forma siguiente. Primero, se comenta la importancia de los laboratorios virtuales para el control de robots. La sección 3 está dedicada a explicar las principales características del laboratorio virtual desarrollado. Posteriormente se describirá el laboratorio remoto. Y finalmente, se comentan algunas conclusiones y posibles trabajos futuros.

2. Los laboratorios virtuales en la robótica.

Los robots se emplean en un amplio rango de aplicaciones, desde fabricación [12] hasta cirugía [10], en tareas como el manejo de materiales peligrosos [9], etc. Actualmente, los dispositivos robóticos permiten incrementar la productividad y flexibilidad, siendo cada vez más utilizados en la industria. Por consiguiente, es fundamental comprender su funcionamiento y tener gente preparada para manejarlos adecuadamente.

La creación y uso de entornos virtuales en la enseñanza puede proporcionar tanto al estudiante

como al operario un adecuado aprendizaje para la utilización de robots. Estas aplicaciones simulan los dispositivos robóticos, con la posibilidad de cambiar todos sus parámetros y el entorno que le rodea. De esta manera, el uso de los laboratorios virtuales en el campo de la robótica ofrece una nueva solución para que tanto el estudiante como el operario adquieran rápidamente conocimientos y habilidades para controlar un robot. Además, gracias al desarrollo de Internet y su uso como medio de comunicación para la enseñanza y adiestramiento [5], el usuario puede ejecutar el laboratorio virtual desde cualquier lugar y en cualquier momento.

Hoy en día, existen muchos grupos de investigación que trabajan en el campo de la realidad virtual. Consecuentemente, existen bastantes laboratorios virtuales desarrollados en el campo de la robótica [2]. Este hecho pone de manifiesto la importancia que tienen dichas aplicaciones, tanto para la enseñanza como para la industria.

A pesar de la cantidad de herramientas que existen, es necesario resaltar que las nuevas aplicaciones que se presentan proporcionan una serie de importantes características tales como: están basadas en Java, ofrecen una arquitectura abierta y flexible al usuario y es posible modificar e interactuar con multitud de parámetros en el dispositivo robótico virtual. Con la realización de estas herramientas, los autores intentan mejorar un laboratorio virtual y remoto que se desarrolló en su grupo de investigación denominado Robolab [3], cuyo éxito en la enseñanza ha sido probado [11].

3. El laboratorio virtual desarrollado

Tal y como se comentó anteriormente, los laboratorios virtuales son herramientas que dan la posibilidad de experimentar y explorar el comportamiento del sistema robótico. De esta manera, ayudan al usuario a comprender complejos conceptos a través de la simulación. Sin embargo, crear un laboratorio virtual requiere tiempo y esfuerzo, debido a que es necesario la descripción del sistema en lenguaje de programación. Mediante Ejs, esta tarea de desarrollo se simplifica.

Ejs es una herramienta *open-source* desarrollada en Java para la creación de

laboratorios virtuales interactivos [7]. Fue parte del proyecto *Open Source Physics*, que se estableció con el objetivo de crear y distribuir simulaciones para su uso pedagógico en la enseñanza de la física [4]. Las simulaciones creadas con Ejs pueden ser usadas como programas *stand-alone* bajo diferentes sistemas operativos o ser distribuidas como *applets* a través de Internet.

La principal razón de escoger Ejs para el desarrollo tanto del laboratorio virtual como del remoto, es su simplicidad de uso y que no es necesario ser un programador experto para crear una aplicación [6]. Por lo tanto, no ha sido necesario emplear una gran cantidad de tiempo para crear una interfaz gráfica interactiva y ha permitido concentrarse más en la descripción del sistema robótico.

Para el desarrollo del modelo del robot del laboratorio virtual se utiliza la conexión entre Ejs y Matlab [8]. El modelo matemático del sistema se define por una serie de archivos en código Matlab (archivos *.m*). Por lo tanto, Ejs sólo se emplea para el desarrollo de la interfaz y para declarar las variables. Dichas variables se conectan con las variables de salida de los archivos Matlab. Así, cada cambio en la interfaz de usuario se envía al *workspace* de Matlab y es leído por el modelo. Al mismo tiempo, las variables del modelo son escritas en el *workspace* que lee Ejs para refrescar la vista. Así, el *workspace* de Matlab es el *buffer* que Ejs y Matlab usan para intercambiar datos (Figura 2).

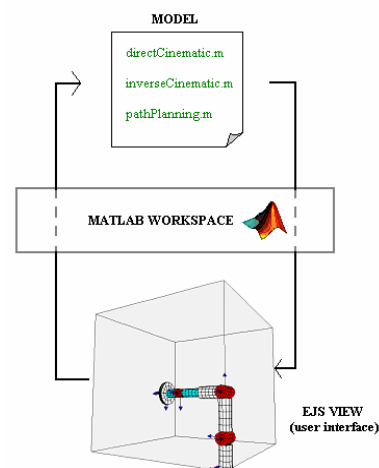


Figura 1. Flujo de datos entre Ejs y Matlab

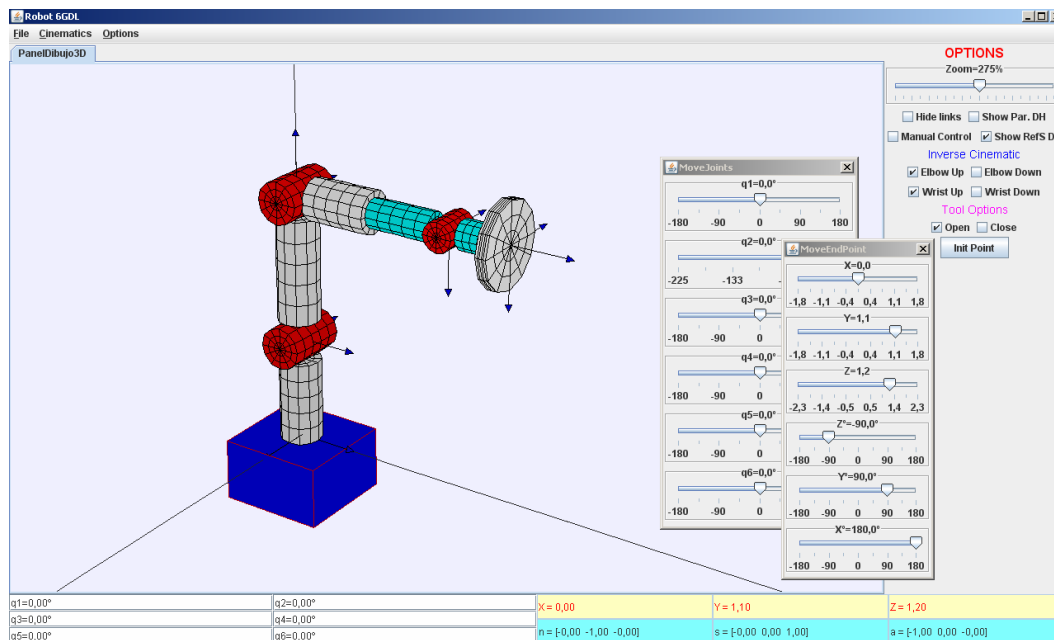


Figura 2. Interfaz de usuario del laboratorio virtual

La vista principal del laboratorio virtual se muestra en la Figura 2. La parte superior contiene una representación 3D del espacio de trabajo donde se visualiza el robot de seis grados de libertad. Este es el lugar de la interfaz donde se representa el movimiento del robot. Podemos observar también, una serie de diálogos que contienen controles mediante los que se mueve el robot (ver sección 3.1). Finalmente, en la parte derecha de la vista podemos ver que hay una serie de controles con más opciones como: zoom, mostrar/ocultar los eslabones y mostrar/ocultar los sistemas de referencia Denavit-Hartenberg. En las siguientes subsecciones se describirán las distintas posibilidades del laboratorio virtual.

3.1. Cinemática directa e inversa

El movimiento del robot se basa en la implementación del modelo cinemático directo e inverso del mismo [8]. Dentro de esta opción, es posible mover el robot mediante controles articulares dando valores a cada una de las articulaciones del robot (controles $q1...q6$), o mediante controles cartesianos dando valores al

punto final (controles X , Y , Z , X° , Y° , Z°). La interfaz muestra en todo momento la matriz de transformación del extremo del robot y los valores de posición de las articulaciones.

3.2. Parámetros geométricos

En el laboratorio virtual es posible editar y cambiar los valores que definen la geometría del robot. Existe un diálogo específico con varios controles para realizarlo. Además de los parámetros geométricos, también es posible cambiar el radio de los eslabones. La simulación gráfica se actualiza dinámicamente mientras el usuario va cambiando los parámetros.

3.3. Planificador de trayectorias

El modelo del sistema implementa un planificador de trayectorias. En concreto, se usa la trayectoria polinómica 4-3-4, que genera un movimiento suave del robot. Este interpolador divide la trayectoria en tres partes: aceleración del actuador, tiempo de máxima velocidad y deceleración del actuador.

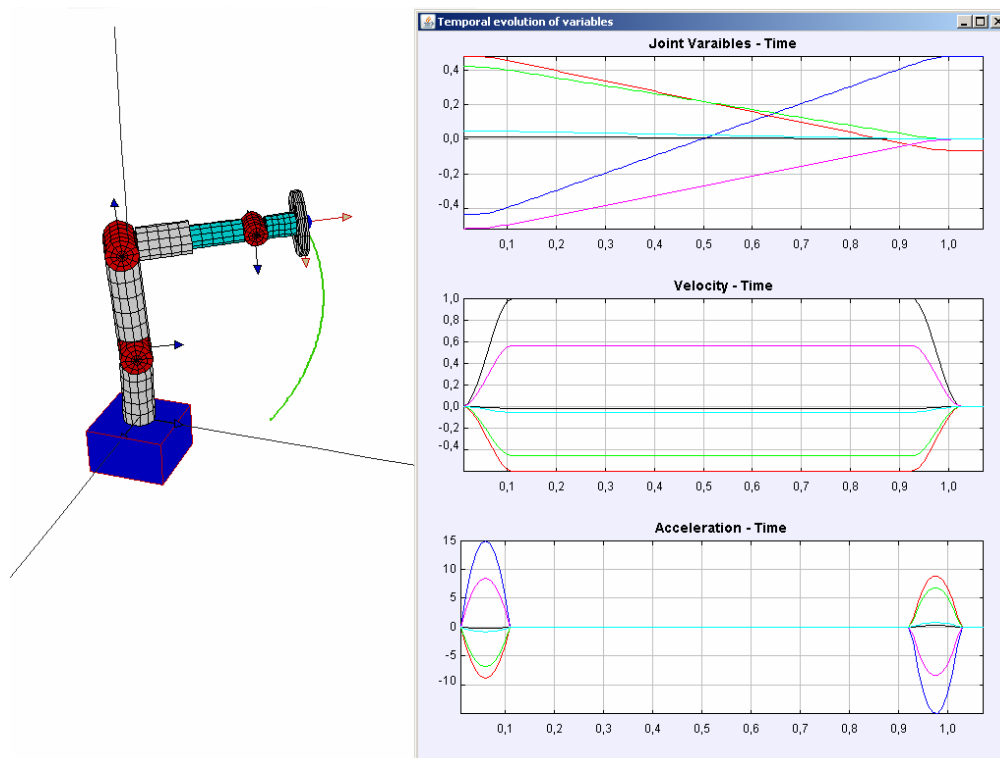


Figura 3. Planificador de trayectorias

Para configurar una trayectoria, el usuario debe introducir los siguientes valores: tiempo de aceleración y deceleración, velocidades articulares máximas y el punto inicial y final. Con respecto a los valores del punto inicial y final, es posible introducirlos tanto en coordenadas cartesianas como articulares. Además, la aplicación grafica la trayectoria 3D en el espacio de trabajo, la evolución temporal de las posiciones, velocidades y aceleraciones articulares (Figura 6).

4. El laboratorio remoto

A parte del laboratorio virtual desarrollado, se ha creado un laboratorio remoto que permite la simulación y teleoperación de un robot tipo Scorbot ER-IX (de Intelitek).

El usuario, mediante un applet desarrollado con Ejs, es capaz de simular y teleoperar un robot a través de Internet, con sólo teniendo instalado en su PC la máquina virtual de Java (*Java Virtual Machine*) y un explorador web. A continuación vamos a describir las características principales del laboratorio remoto.

4.1. Simulación

El aspecto gráfico de la interfaz se muestra en la Figura 4. Como podemos observar con respecto al laboratorio virtual anterior, las texturas de los objetos 3D son diferentes. Este hecho se debe a que se ha utilizado una versión nueva de Ejs, que implementa un panel gráfico basado en objetos 3D de la API de Java.

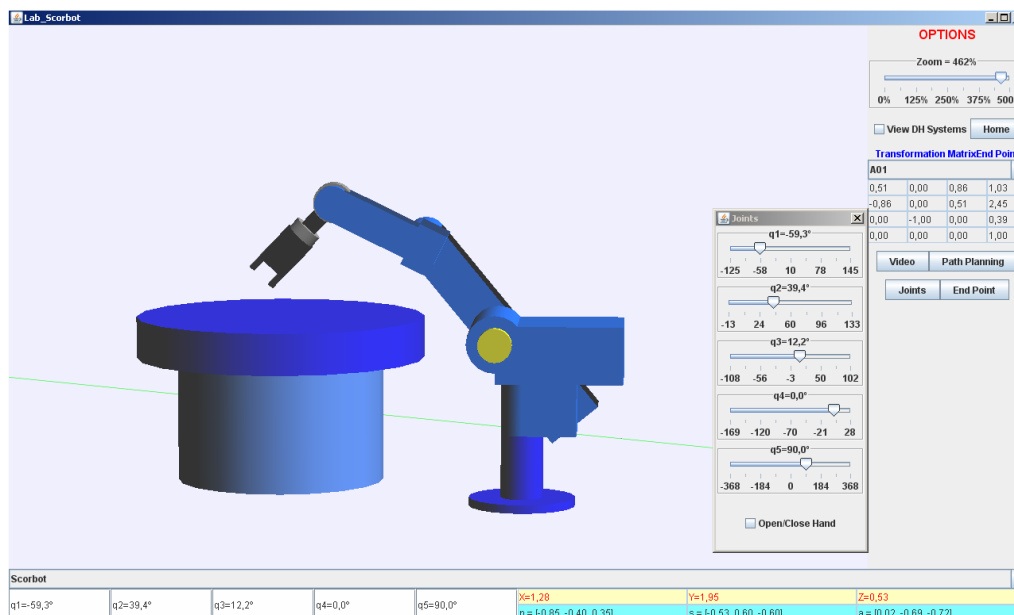


Figura 4. Interfaz de usuario del laboratorio remoto

La aplicación permite al usuario practicar conceptos como la cinemática directa, inversa y planificación de trayectorias. Para esta última opción, la aplicación permite al usuario simular trayectorias articulares de tipo *síncrono* y *asíncrono*. Además, estas trayectorias se pueden guardar en una lista y ejecutarlas todas a la vez creando una trayectoria formada por varios puntos intermedios.

4.2. Teleoperación

Una vez que el usuario ha obtenido una serie de trayectorias válidas, puede solicitar la ejecución remota de la lista completa de comandos en el robot real. Dichos comandos son enviados *servidor del robot*, PC que se encarga de comprobar son válidos antes de trasladarlos al lenguaje del robot. Esta acción garantiza el buen uso del sistema real.

Para que el usuario pueda seguir el movimiento real del robot, la aplicación dispone de dos opciones de realimentación [1]. La primera es mediante un flujo de video comprimido que se transmite de forma on-line al ordenador del usuario y mediante una actualización de la

simulación con información on-line recibida del controlador del robot de sus posiciones articulares reales (Figura 5).

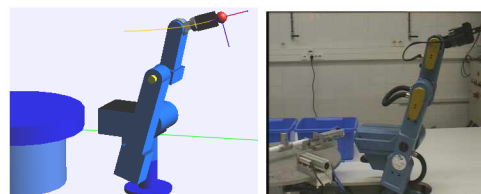


Figura 5. Opciones de realimentación

5. Experiencia con Ejs

En este punto vamos a comentar los principales resultados obtenidos del empleo de Ejs como herramienta para el desarrollo de las aplicaciones. La creación de un laboratorio virtual con Ejs es muy simple. Los autores han empleado poco tiempo en desarrollar los presentados aquí. La interfaz obtenida es muy similar a las de otras aplicaciones desarrolladas en Ejs (Figura 5). Además, las herramientas desarrolladas pueden ser integradas con otras creadas con Ejs en un

entorno como *eMersion*, herramienta de aprendizaje que se encarga de la gestión de este tipo de simulaciones.

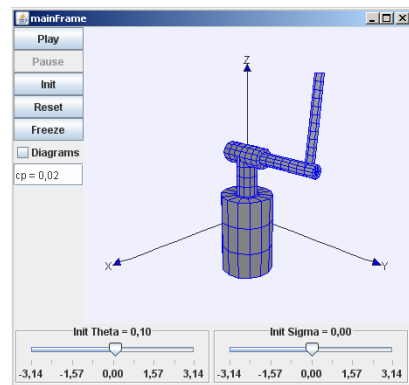


Figura 6. Ejemplo de interfaz de Ejs

6. Trabajos en curso

Actualmente, los autores continúan desarrollando las aplicaciones descritas en este artículo, incluyendo nuevas características tanto en la interfaz gráfica como en el modelo del sistema robótico simulado.

Con respecto a la interfaz, se está trabajando en la posibilidad de añadir nuevos robots de diferentes características en el espacio de trabajo, ya que las aplicaciones actuales sólo contienen un robot. De esta forma, el usuario podrá practicar con diferentes tipos de robots en un entorno más real. También, se está trabajando en mejorar el aspecto gráfico 3D de la versión actual de Ejs. Así, se conseguirá más realismo en las simulaciones.

Con respecto al sistema robótico, la principal característica es añadir el modelo dinámico. El sistema actual sólo incluye la cinemática directa e inversa y el planificador de trayectorias. Dada la complejidad de este trabajo, se recurrirá a la comunicación que posee Ejs con softwares orientados al modelado de sistemas, tales como *Modelica* o *Matlab*. Con el modelo dinámico, la aplicación será mucho más completa y el usuario podrá evaluar el par en los actuadores cuando el robot simula una determinada tarea.

Posteriormente a las características comentadas, los autores trabajarán en añadir otras

opciones como: reconocimiento 3D de objetos básicos en el entorno de trabajo, la inclusión de objetos virtuales usando realidad aumentada y un intérprete de comandos de voz para que el usuario guíe la tarea.

7. Conclusiones

En este artículo se ha presentado un laboratorio virtual y otro remoto para la simulación y teleoperación de robots industriales. Nuestras aplicaciones están principalmente orientadas a la enseñanza y práctica de la robótica. El empleo de los laboratorios virtuales en el campo de la educación y aprendizaje permite tanto al alumno como al operario ser adiestrado de una manera rápida y económica. De esta manera, estarán más preparados para manipular y resolver los problemas de los complejos dispositivos robóticos.

Las aplicaciones se han desarrollado usando Ejs, un *open-source* software diseñado para la creación de simulaciones interactivas. Mediante esta herramienta, el procedimiento para transformar el sistema robótico en un interactivo laboratorio virtual fue más sencillo que con cualquier otro programa. También es importante comentar que no se ha empleado mucho tiempo ni se ha necesitado adquirir habilidades de programación para realizar la interfaz gráfica de las simulaciones.

Con las aplicaciones creadas, el alumno puede aprender conceptos de robótica tales como la cinemática directa e inversa y planificación de trayectorias. La interfaz de usuario y la simulación gráfica son muy reales y aptas para cualquier persona. Además, Ejs genera el *applet* de la simulación y con ello, la posibilidad de poder ejecutar la aplicación desde otros lugares en cualquier momento.

Actualmente, los autores están ampliando las capacidades de las herramientas en diferentes aspectos para lograr potentes aplicaciones.

Agradecimientos

El trabajo presentado en este artículo está financiado por el Ministerio de Educación y Ciencia a través del programa de becas FPI y el proyecto de investigación DPI2005-06222. Los autores quieren agradecer esta financiación.

Referencias

- [1] Candelas, F. Flexibe system for simulating and teleoperating robots through the Internet. *Journal of Robotic Systems*, 2005.
- [2] Candelas, F. Recursos didácticos basados en Internet para el apoyo a la enseñanza de materias del área de Ingeniería de Sistemas y Automática. *Revista Iberoamericana de Automática e Informática Industrial*, 2005.
- [3] Candelas, F. A., Puente, S. T., Torres, F., Ortiz, F. G., Gil, P., Pomares, J. A virtual laboratory for teaching robotics. *International Journal of Engineering Education*, 2003.
- [4] Christian, W., Belloni, M. Developing open source programs for science and mathematics. *EUROCON* 2003.
- [5] Dormido, S. Control Learning: Present and Future. *Annual Reviews in Control*, 2004.
- [6] Dormido, S., Farias, G., Sanchez, J., Esquembre, F. Adding interactivity to existing Simulink models using Easy Java Simulations. *European Control Conference on Decision and Control*, 2005.
- [7] Esquembre, F. Easy Java Simulations: a software tool to create scientific simulations in Java. *Computer Physics Communications*, 2004.
- [8] Jara, C. A. Diseño de herramientas para la interacción remota on-line de robótica industrial. *Memoria de suficiencia investigadora*, Alicante, 2007.
- [9] Sabater, J. M., Saltarén, R. J., Aracil, R., Yime, E., Azorín, J. M. Teleoperated parallel climbing robots in nuclear installations. *Industrial Robot: An International Journal*, 2006.
- [10] Thorne, G. C., Halliwell, M., Wells, P. N. T. An experimental telesurgery robot. *IEEE Colloquium on Towards Telesurgery*, 1995.
- [11] Torres, F., Candelas, F. A., Puente, S. T., Pomares, J., Gil, P., Ortiz, F. G. Experiences with virtual environment and remote laboratory for teaching and learning robotics at the University of Alicante. *International Journal of Engineering Education*, 2006.
- [12] Vazquez, A. Distributed control system of an experimental robotic cell with 3D vision. *International Conference on Informatics in Control (ICINCO)*, 2006.